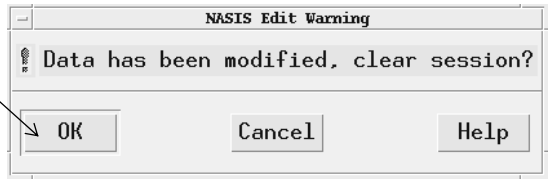
Chapter 17 Writing Custom Queries

The primary way to get data into the selected set is to run a query, and though several prewritten queries come with NASIS, you will likely want to write some of your own. In this lesson, you will learn to write a basic query that demonstrates the significance of target tables.

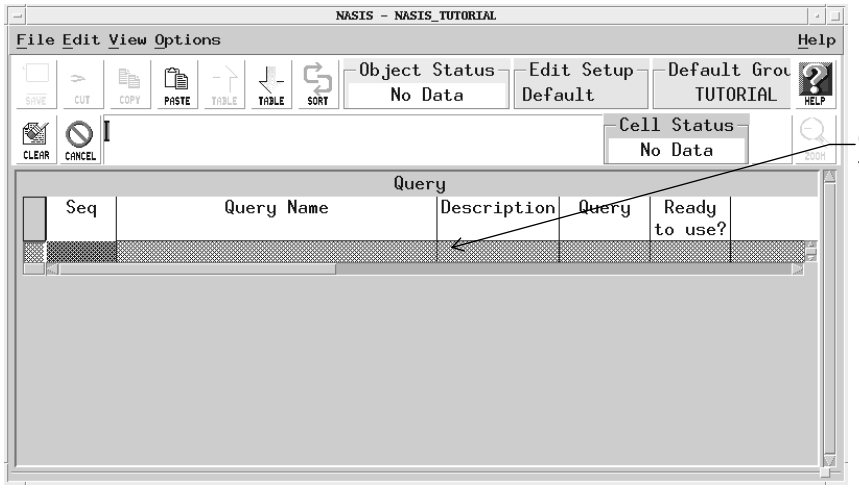
A query has ownership the same way data mapunits, legends and area types have ownership. Like other owned objects, queries are not loaded until you run a query that retrieves them. You can create, save, and share queries.

Loading the Query Table

1. Before you begin this lesson, clear the selected set by selecting the **File** menu, then clicking **New**. Click **OK** if the following message appears.



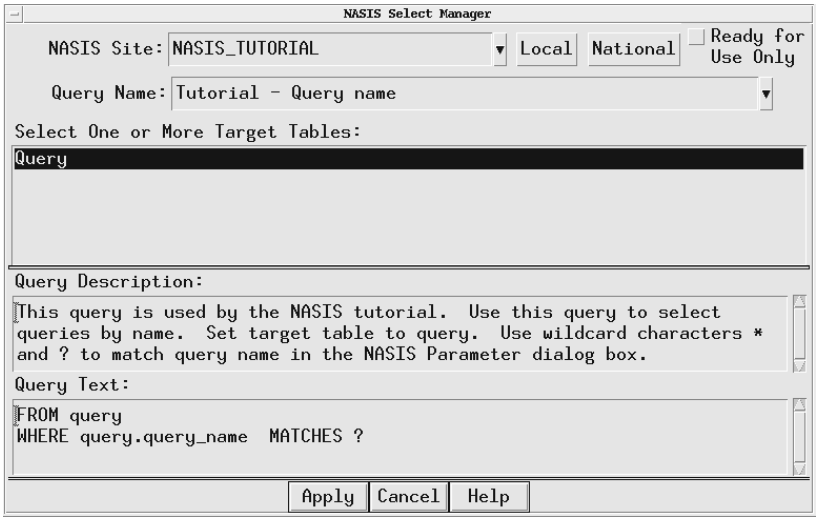
Open the Query table by selecting **View** menu, **Queries**, then clicking **Query**. The Query table is empty until you run a query to load the table.



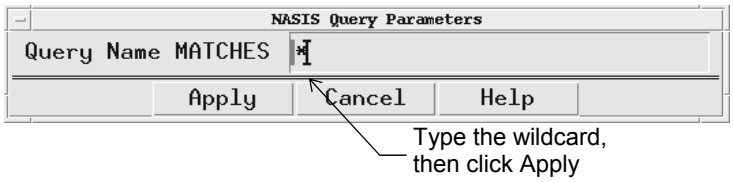
2. You need to retrieve the queries from the database by running a query. On the **File** menu, choose **Select** to open the Select Manager.

   Note: If you only want to view—not edit—queries, you can see them all from the Select Manager. But to edit queries, you must load them into the Query table.

3. Select the query called **Tutorial - Query Name**.

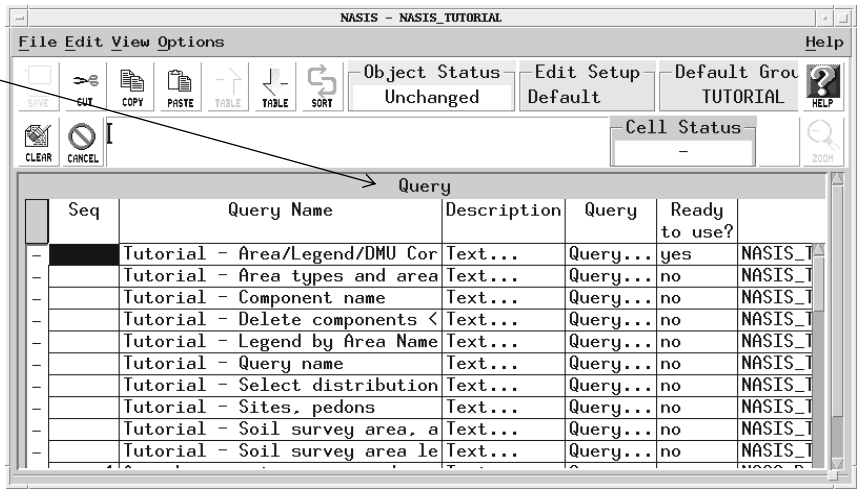


4. Read the **Query Description** to get an idea of what this query is designed to do and how to run it.
5. Click **Apply**.
6. To retrieve all queries, type the wildcard "*****" in the parameter box, then click **Apply**.



8. A message reports that fifty-one queries were added to the selected set. You now have in your selected set a copy of the fifty-one queries in the tutorial database. Click **OK**.
9. Click **Cancel**. You can see that the Query table is now populated. The Query table stores the name of a query, a description of it, its ownership data, and the actual query language.

Query table with all queries in the tutorial database

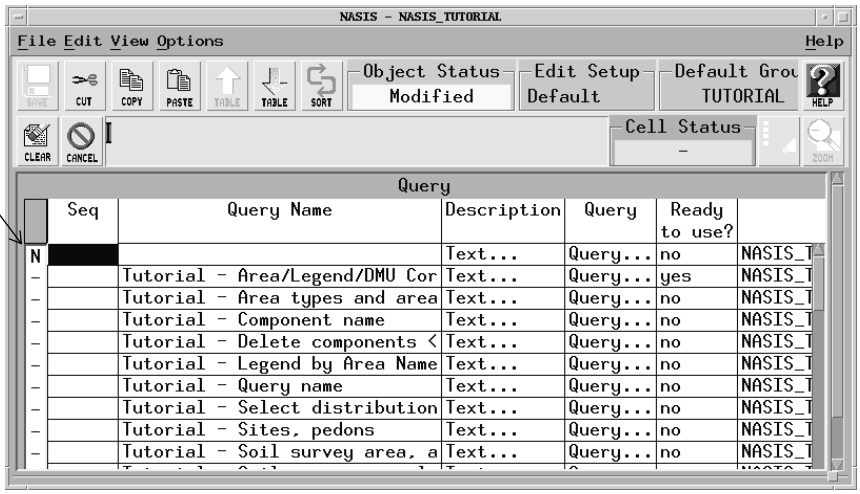


Writing a New Query

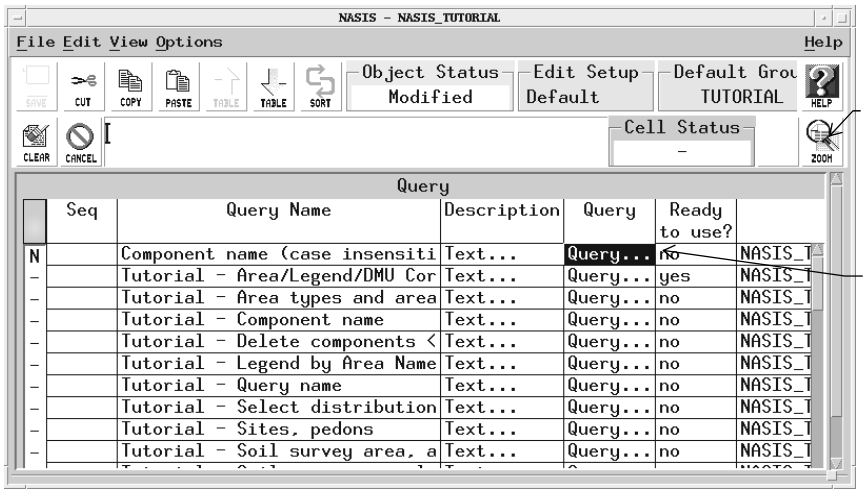
You want to write a query to load into your selected set all Shapleigh components in your dataset so you can view and possibly edit them. Taking it a step further, you decide to write a query that allows you the flexibility, in the future, to retrieve *any* component.

1. In the Query table, insert a new row by pressing F8 (or select **Edit** menu, then choose **Insert**).

New row in the Query table

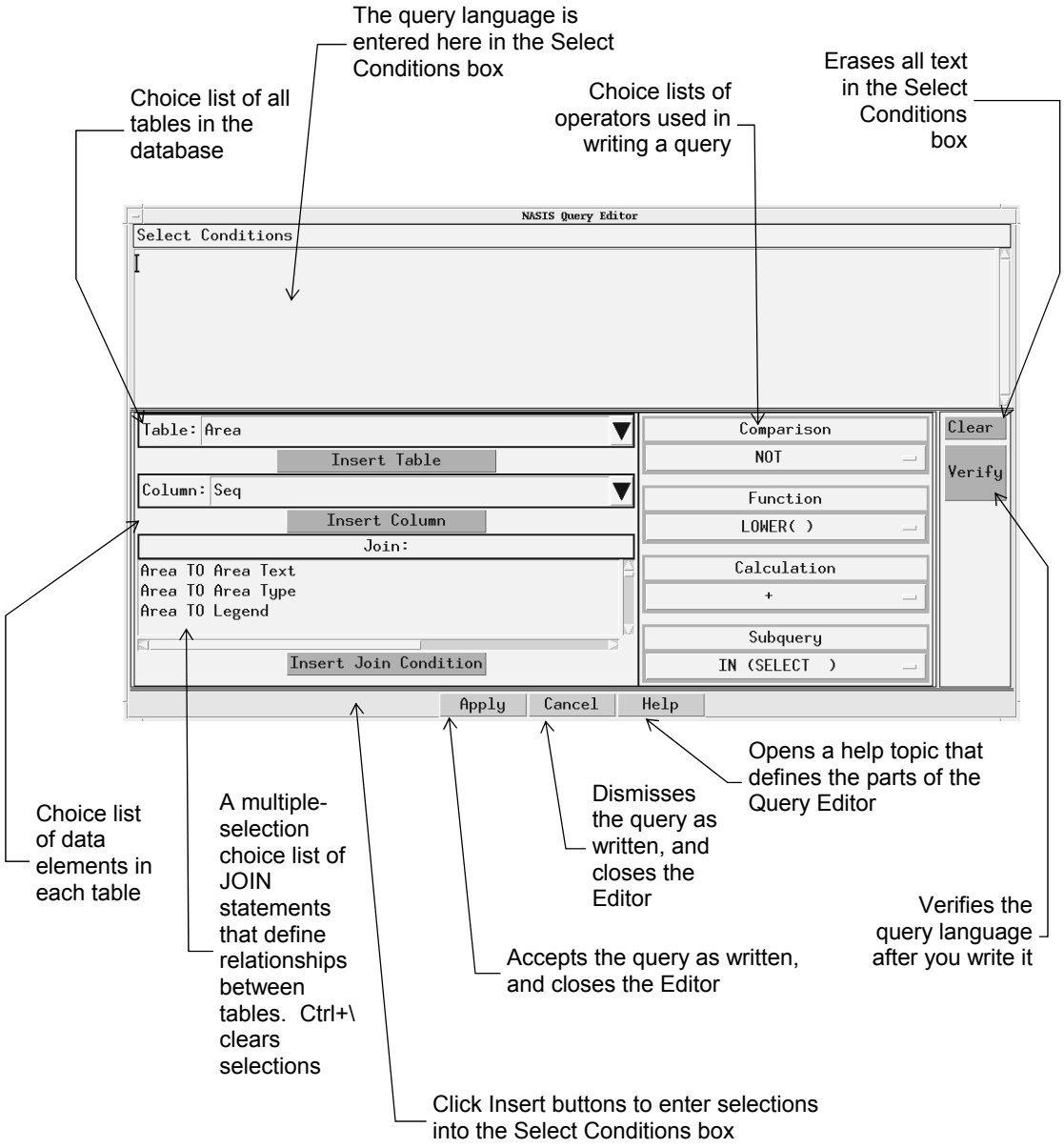


2. In the Query Name field, type **Component name (case insensitive)**.
3. Skip the **Description** field for now, and highlight the **Query...** field.
4. Click the **Zoom** button to open the Query Editor—a specialized editor for writing NASIS queries.



5. The NASIS Query Editor appears. Before you write the query, take a moment to examine the parts of this editor. (See the next page.)

   Note: The query language has two parts: FROM clause and WHERE clause. The Select Conditions box (where the cursor is blinking) is where the query language is entered. The behavior of the Select Conditions box is like a typical word processor.



6. Refer to "Appendix B. Query Tools, Rules, Samples" in the back of this manual, and follow the methodology for writing queries (pages B.1-3). Also read through the rules on query writing (pages B.3-4) and the explanation of how to identify data types (pages B.5-6).
7. To begin writing a query, ask yourself, what data element am I looking for? In this case, you are looking for Component Name. What table is the element in? The Component table.

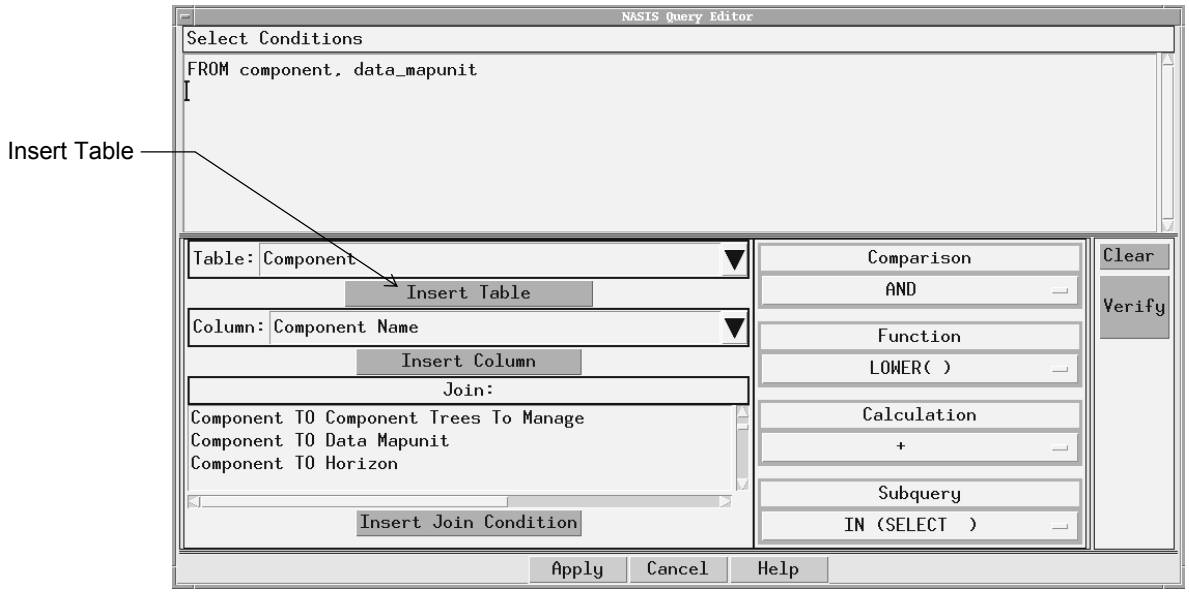
8. In the Select Conditions, type **FROM** and press the **SPACEBAR**. (The Query Editor is not case-sensitive, so you could enter "from" in lowercase letters.)

   Note: The FROM clause includes all tables you might choose as a target table. Target tables focus the query. To add flexibility to your query, and for demonstration purposes, you will add more than one table here.

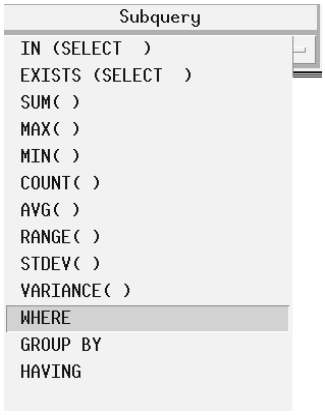
9. In the **Table** field, click the black triangle button. A choice list displays all NASIS tables.
10. On the choice list, click **Component**. The choice list closes automatically.
11. Click the **Insert Table** button. The Component table name is inserted into the FROM clause in the Select Conditions box.
12. After *component*, type a comma.
13. In the **Table** field, open the choice list, scroll down to **Data Mapunit** and click it.

    Note: You may be wondering why you would target the Data Mapunit table when the purpose of the query is to load components one at a time. Entering Data Mapunit here lets you later choose it as a target table. It makes your query more flexible. It will also let us demonstrate the difference between using Component as the target table and Data Mapunit as the target table.

14. Click the **Insert Table** button. The Data Mapunit table name is inserted.

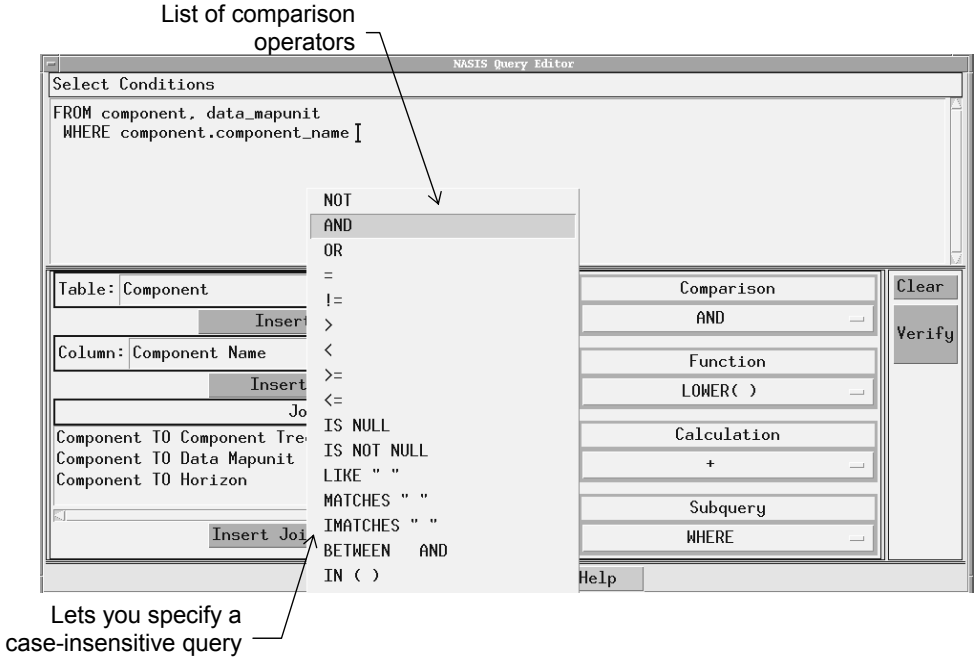


15. On a separate line, type **WHERE** and the **SPACEBAR** key. Or, select WHERE from the subquery list.



Note: The WHERE clause specifies the conditions to be placed on data elements. First, you will insert the data elements into the WHERE clause.

16. In the **Table** field, open the choice list and select **Component**. But this time, don't insert the table. By selecting the table, you narrow the column field to only the columns in a given table.
17. In the Column field, click the black triangle button. From the choice list, select **Component Name**.
18. Click the **Insert Column** button. The data element *component name* (including its table name) is inserted into the WHERE clause.
19. With the data element entered, you now need to place conditions on it. In this new query, you want to retrieve specified components. Click the **Comparison** operator choice list and look at the options.



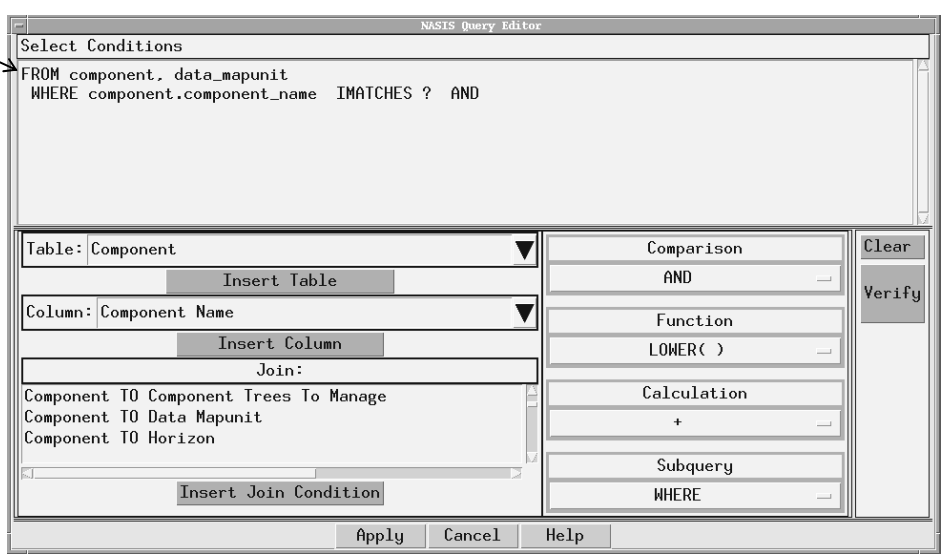
Note: The use of comparison operators is sometimes limited by data types. Next, you will identify the data type of component name.

20. On the Query Editor window, click the **Help** button.
21. On the button bar, click **Help Topics**.
22. On the **Contents** tab, click **References**, and then click the **Open** button in the lower right corner.
23. Click on **Database Structure and Elements**, and then click the **Open** button.
24. Click on the **Data elements and data types** topic, then click the **Display** button at the bottom to view the **Data elements and data types** PDF file.

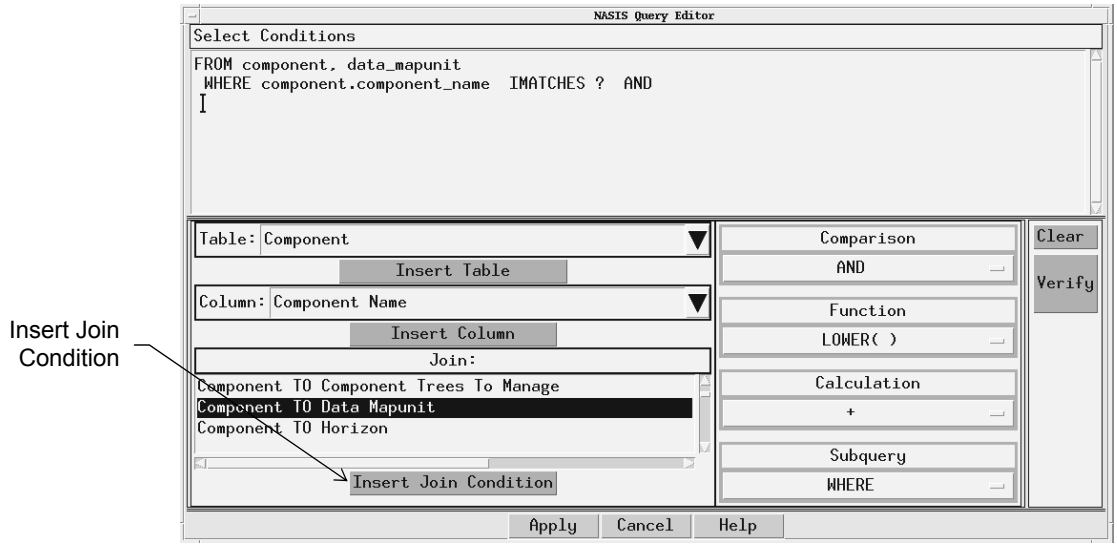
    Note: When you clicked the link, the system began loading the *Adobe Acrobat Reader*. This process takes a few seconds. Do not click again, or it will start the reader more than once. The reader will open with the Data elements and data types document displayed.
25. In the list of tables, click the **Component** table. Your cursor will move to the page containing elements for the component table. Move to the row containing the **Component Name**. In the far right column, identify the **Data Type,** it is a variable character (var char).
26. Next, you want to find the valid comparison operators for use with variable character data elements. Close the *Acrobat Reader*.
27. On the Help Topics dialog, open the **Data types and comparison operators** topic. This chart is also shown on page 6 of Appendix B.
28. Enlarge the screen if necessary, and scroll down to **variable character**. By reading across the table, you find that any of the comparison operators are valid operators for use with variable character data elements.
29. Finally, ask yourself: What comparison operator will meet the conditions? Three are likely possibilities: equals (=), MATCHES, or IMATCHES.

    Note: If (=) is used, the component name must be typed exactly as it was entered into the database. If MATCHES or IMATCHES are used, the (*) and (?) are allowed to replace one (?) or more (*) characters. If you're not sure if the component name is upper case, lower case, or a combination of both, use IMATCHES.
30. Reduce the help window to an icon, and return to the NASIS Query Editor.
31. On the **Comparison** operator choice list, select **IMATCHES**.

    Note: Refer to "Appendix B. Query Tools, Rules, Samples" for more information on comparison operators.
32. The query requires that you decide whether or not to use parameter substitution. You ask yourself: Does the data type allow for parameter substitution, and do I want flexibility at run-time? Because you want flexibility to choose a specified component to retrieve, delete the quotation marks that follow IMATCHES, and type a question mark (?).
33. When multiple tables exist in the FROM clause, you must define the relationship between them by adding JOIN statements to the WHERE clause. Type **AND** then start a new line.



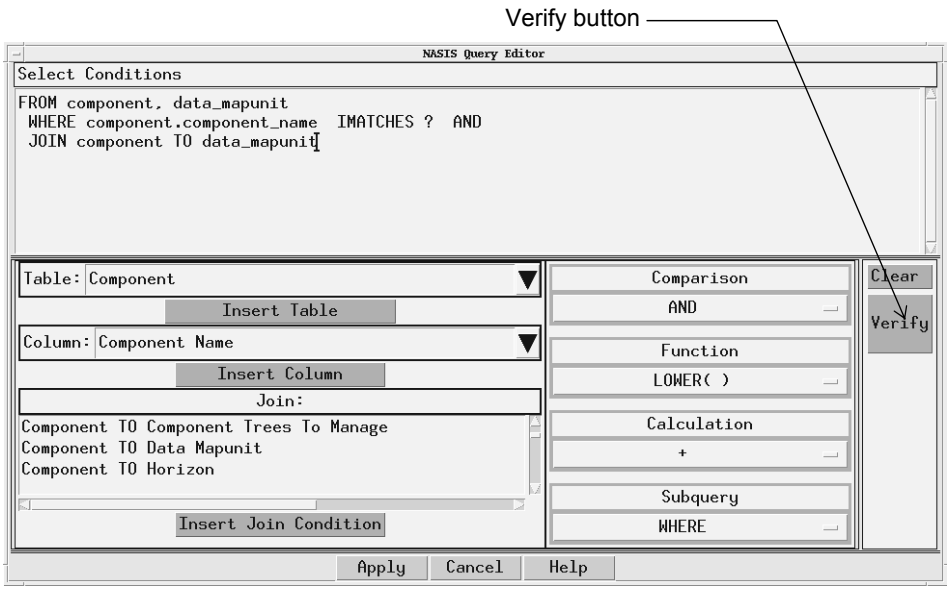
34. In the Join field, scroll to **Component to Data Mapunit**, and click it.
35. Click **Insert Join Condition**.



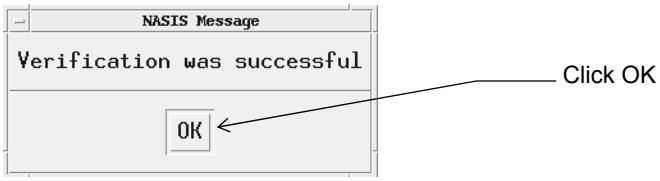
Note: If you would scroll farther down in the list of joins, you would find *Data Mapunit to Component*. The JOIN Statement produces the same results whether you define the table relationship as *Component to Data Mapunit* or *Data Mapunit to Component*.

Note: For future reference, this is a multiple-selection choice list. To simultaneously deselect all the selections in the choice list, type **CTRL**+\. To deselect an individual selection, click it.

36. You have finished writing the query language. Verify the syntax by clicking the **Verify** button.



35. A message should appear indicating that the verification was successful. Click **OK**.



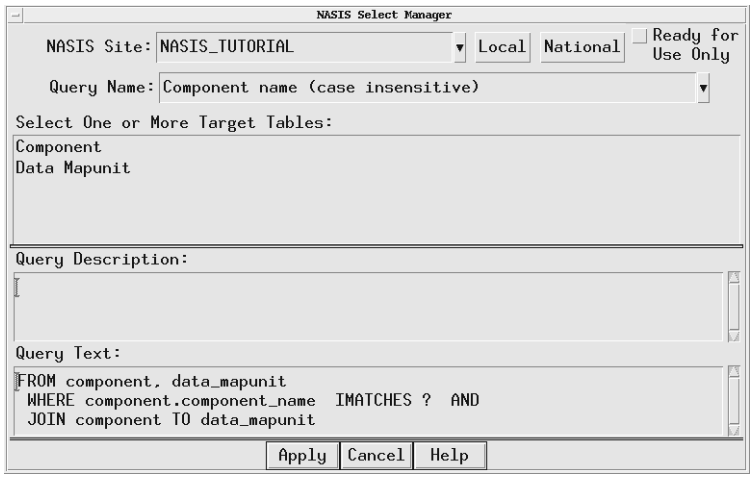
36. Accept the query as written and close the Query Editor by clicking **Apply**. You are returned to the Query table.
37. Before you close the Query table, highlight the **Query...** field and click the **Zoom** button.

    Note: Notice that the query you wrote still appears in the table. Because you are working in the tutorial database, you cannot save your query. The query will be lost when you clear the selected set or exit NASIS. When you are working in your own database, you will want to save new or modified queries to the permanent database by choosing File–Save or clicking the Save button on the tool bar.
38. **Cancel** to close the Query Editor.

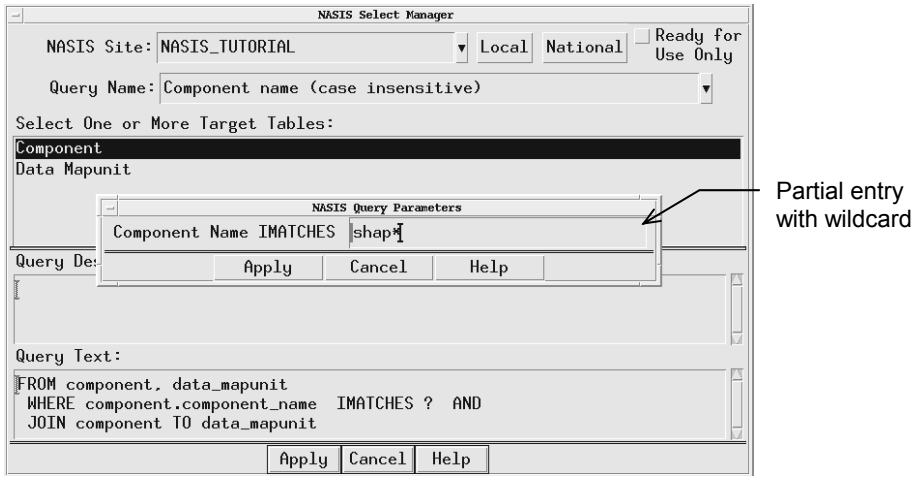
Running the New Query

The records presently in your selected set include the fifty-one queries you loaded at the beginning of this lesson in addition to the new query you just wrote. To get Shapleigh components, you must run the component query and specify Shapleigh.

1. On the **File** menu, choose **Select** to open the Select Manager.

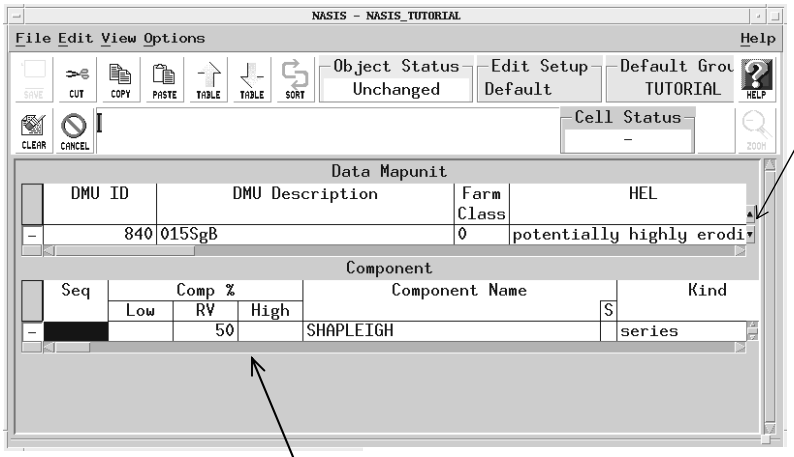


2. The new query should appear, but if not, select it from the Query Name column.
3. Highlight the **Component** target table.
4. Run the query by clicking **Apply**.
5. In the Query Parameters dialog, type **shapleigh** or type a partial entry and use a wildcard; then click **Apply**.



6. A message reports that six components were added to the Component table and six Data Mapunits were added to the Data Mapunit table. Click **OK**.
7. Close the Select Manager by clicking **Cancel**.
8. Look at the components by selecting the **View** menu, **Components**, then clicking **Component**.

   Note: In the Component table, the data mapunit's SHAPLEIGH component is highlighted. The data mapunit 015SgB is the first of six retrieved from the database.



9. In the **Data Mapunit** table, click the down scroll arrow (shown on the previous sample screen) to view each data mapunit and its SHAPLEIGH component.

   Note: The component percentage for Shapleigh in each data mapunit is 50%. Shapleigh is in a complex with another component, but the component query did not load those components into the selected set because you used Component table as the target table.

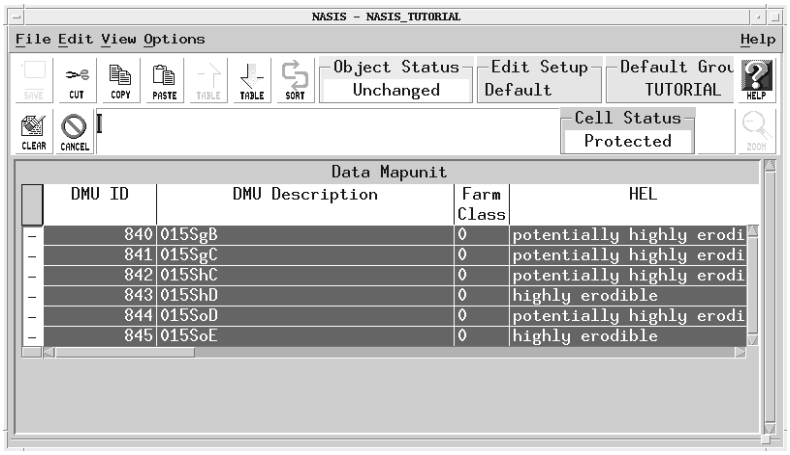
Running the New Query with a Different Target Table

Next, you will change the target table to Data Mapunit table so that in addition to loading the data mapunits that have Shapleigh components, your query will also load the other components in those data mapunits. This section will demonstrate the impact of using target tables.

1. With the Component table open, click the **Up table** button.

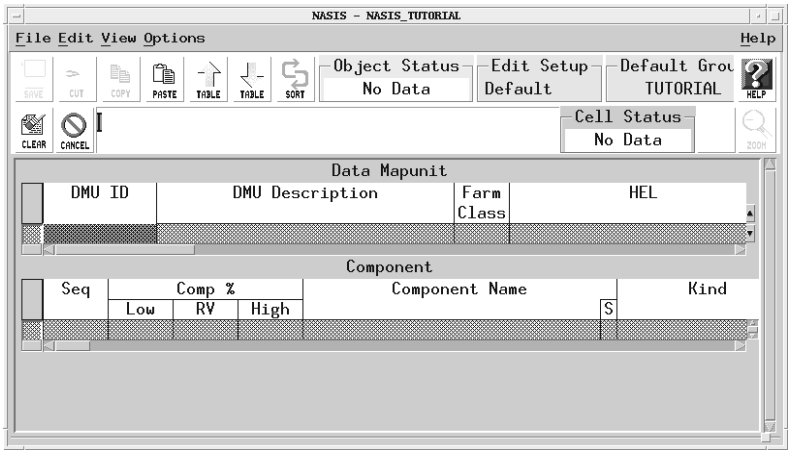
   Note: To fully understand the concept of target tables, you will need to clear the selected components before you run the query again. However, if you were to use the New command, you would also remove the new query you wrote. *New* clears all records in the selected set. Instead, you will use a feature called De-select to remove only the six data mapunits from your selected set.

2. To highlight all six data mapunits, point the cursor in the row status column of the first DMU and click and hold down the mouse button, then drag the pointer down through the table, haighlighting all six DMUs, then release the mouse button.



Note: The row status column is the column to the far left that contains a dashed line. An alternative way of selecting multiple rows is to highlight the first row, then while holding down the CTRL key, click the bottom row.

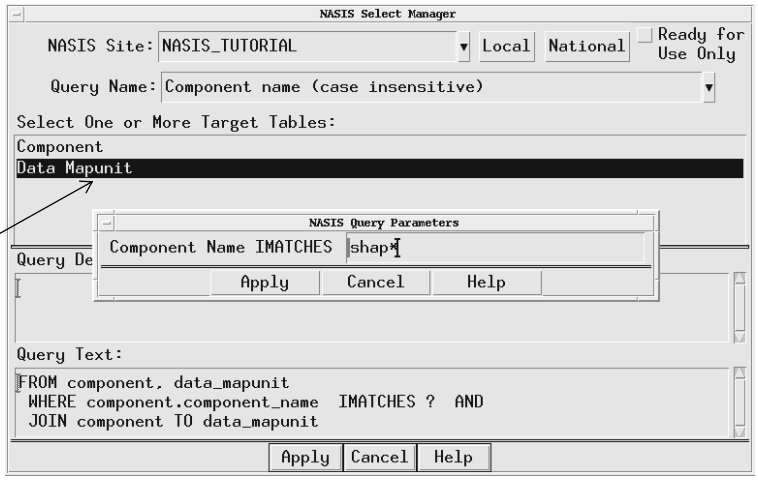
3. On the **File** menu, choose **De-select**.



Note: The Data Mapunit and Component tables become empty.

4. On the **File** menu, choose **Select** to open the Select Manager.
5. Select the same query: Component **name (case insensitive)**.
6. In the Target table field, choose **Data Mapunit**.

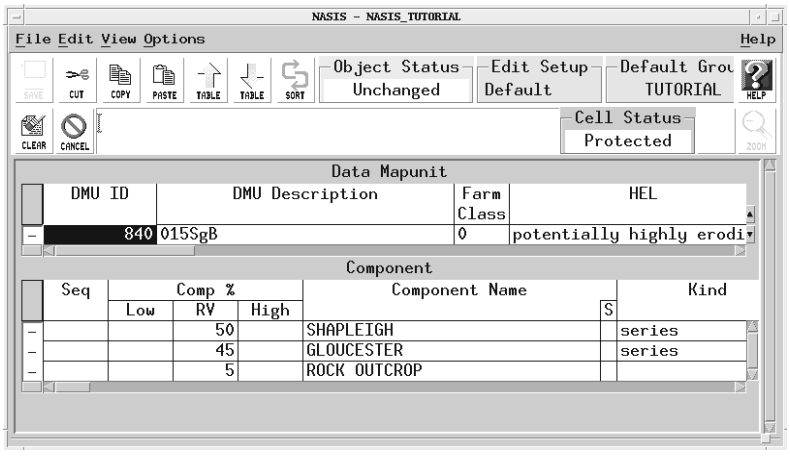
   Note: The Data Mapunit table becomes the focus of the query.
7. Click **Apply**.
8. Type **shap*** then click **Apply**.



Choose only Data Mapunit as the target table

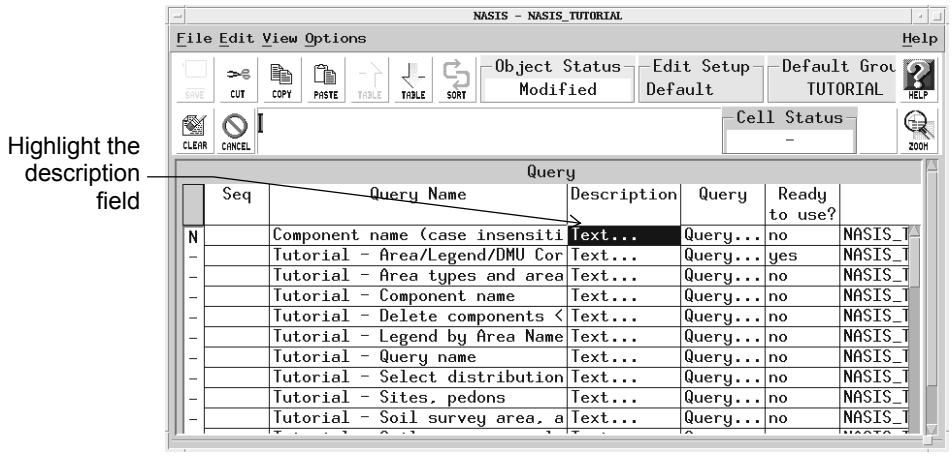
9. This time, a message indicates six rows were added to the Data Mapunit table. Click **OK**.
10. Close the Select Manager by clicking **Cancel**.

    Note: The six data mapunits appear in the Data Mapunit table.
11. Open the Component table (as shown on the following page) by clicking the **Down table** button.



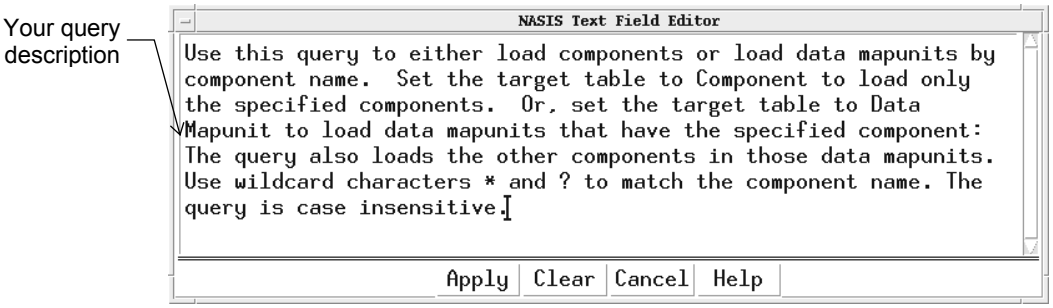
Note: In your selected set, you have *all* components that belong to the data mapunits that have a Shapleigh component. By running the same query but with a different target table, data mapunit, you loaded an entirely different set of records into the selected set. Understanding target tables is essential to using NASIS. Refer to the "Summary of Target Tables" at the end of this chapter.

13. You are almost finished writing your new query. When you are done looking at the records you loaded, select the **View** menu, **Queries**, then click **Query**.
14. You were instructed to skip the Description field when you wrote the query, but click it now.



15. Open the text field editor by clicking the **Zoom** button.

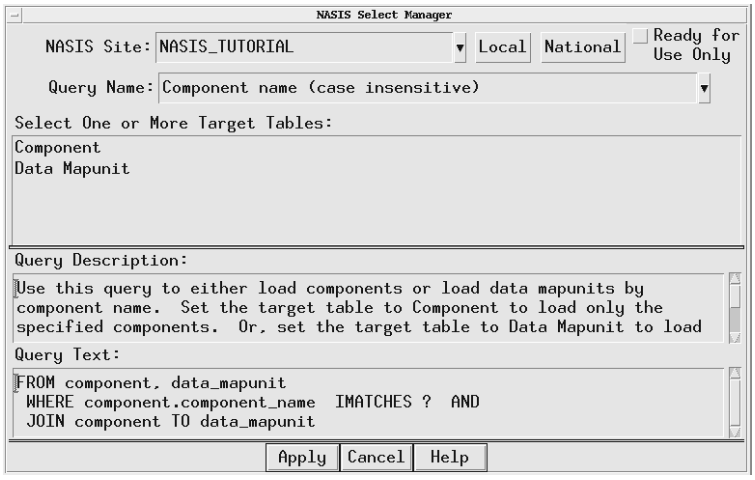
16. The Query Description is an explanation of what the query is designed to do and how to run it. Type a description such as the one in the sample screen below.



17. Close the text field editor by clicking **Apply**.

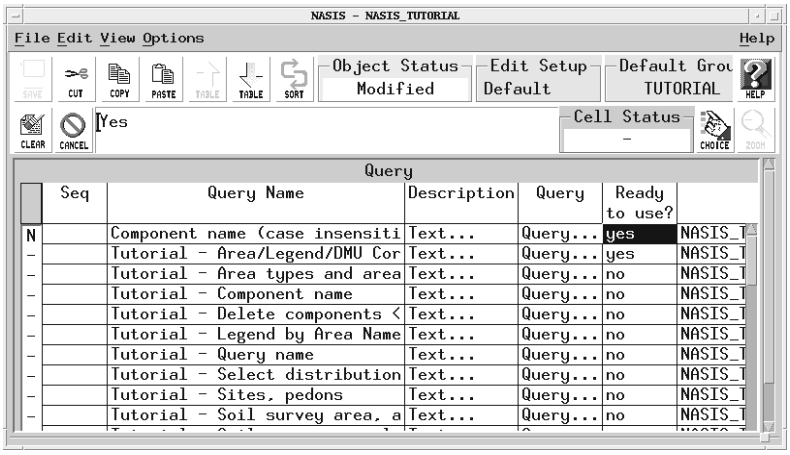
18. Select the **File** menu, and choose **Select**.

    Note: Look at the Query Description of your new query. It now contains the information you typed in the text field editor.



19. If you would like to, spend some time in the Select Manager reading some of the query descriptions. It will give you a better idea of how records can be queried from the database.

20. When you are finished, close the Select Manager by clicking **Cancel**.
21. In the Query table, you have a finished query. If you were working outside the tutorial database, you could save the query and use it to pull specific components into your selected set.



Note: You have finished this lesson. A summary of target tables follows.

Summary of Target Tables

- Simply put, the target table focuses the outcome of a particular query. In this way, you can control the query so that it loads only the specific data you want to work with during an edit session. This is particularly important when you perform global editing operations.
- Some queries are designed to select records from different database objects, for example, from the Legend object and from the Data Mapunit object. You may select on target table per object.
- Your choice of target tables is restricted to those tables that appear in the FROM clause of the query. The first table in the FROM clause is the default target table.
- By entering all possible target tables when you write a query, you give yourself the flexibility at run-time to narrow the focus of the query depending on the actions you intend to perform on the data.
- To fully understand target tables, you need to examine other queries and their outcomes based on the target table. Please refer to "Appendix B. Query Tools, Rules, Samples."